MV/OA 3250 Project Presentation

Spring 2002

K. Mnif

XML Translation Of Armor Information Using XSLT

Table of Contents

XSL - On the Client

A JavaScript Solution

A more versatile solution would be to use a JavaScript to do the XML to XHTML transformation.

By using JavaScript, we can:
- **do browser-specific testing**
- **use different style sheets according to browser and user needs**

XSL transformation on the client side is bound to be a major part of the browsers work tasks in the future, because of the growth in the specialized browser market (Braille, aural browsers, Web printers, handheld devices, etc.)

XML file does not have a reference to the XSL file. XML file could be transformed using many different XSL files.

XSL - On the Client

Here one source code used to transform the XML file to XHTML on the client:

```
<html
>body
>script
type="text/javascript">
// Load XML
var xml = new ActiveXObject("Microsoft.XMLDOM")
xml.async =
false
xml.load("armor.xml")

// Load XSL
var xsl = new ActiveXObject("Microsoft.XMLDOM")
xsl.async =
false
xsl.load("armor.xsl")

// Transform
document.write(xml.transformNode(xsl)
</script
>
</body
</html
>
```

XSL - On the Server

A Cross Browser Solution
-**XSL** can be used to transform a document from **XML** to XHTML in the browser. We let a JavaScript use an XML parser to do the transformation. This solution will not work with a browser that doesn't support an XML parser.
 To make XML data available to all kinds of browsers, we have to transform the XML document on the SERVER and send it as pure XHTML to the BROWSER.

XSL transformation on the server is bound to be a major part of the Internet Information Server work tasks in the future, as we will see a growth in the specialized browser market (Braille, aural browsers, Web printers, handheld devices, etc.)

XSL - On the Server

Here one source code used to transform the XML file to XHTML on the Server:

```
<%
'Load XML
set xml = Server.CreateObject("Microsoft.XMLDOM")
xml.async = false
xml.load(Server.MapPath("armor.xml"))

'Load XSL
set xsl = Server.CreateObject("Microsoft.XMLDOM")
xsl.async = false
xsl.load(Server.MapPath("armor.xsl"))

'Transform File
Response.Write(xml.transformNode(xsl))
%>
```
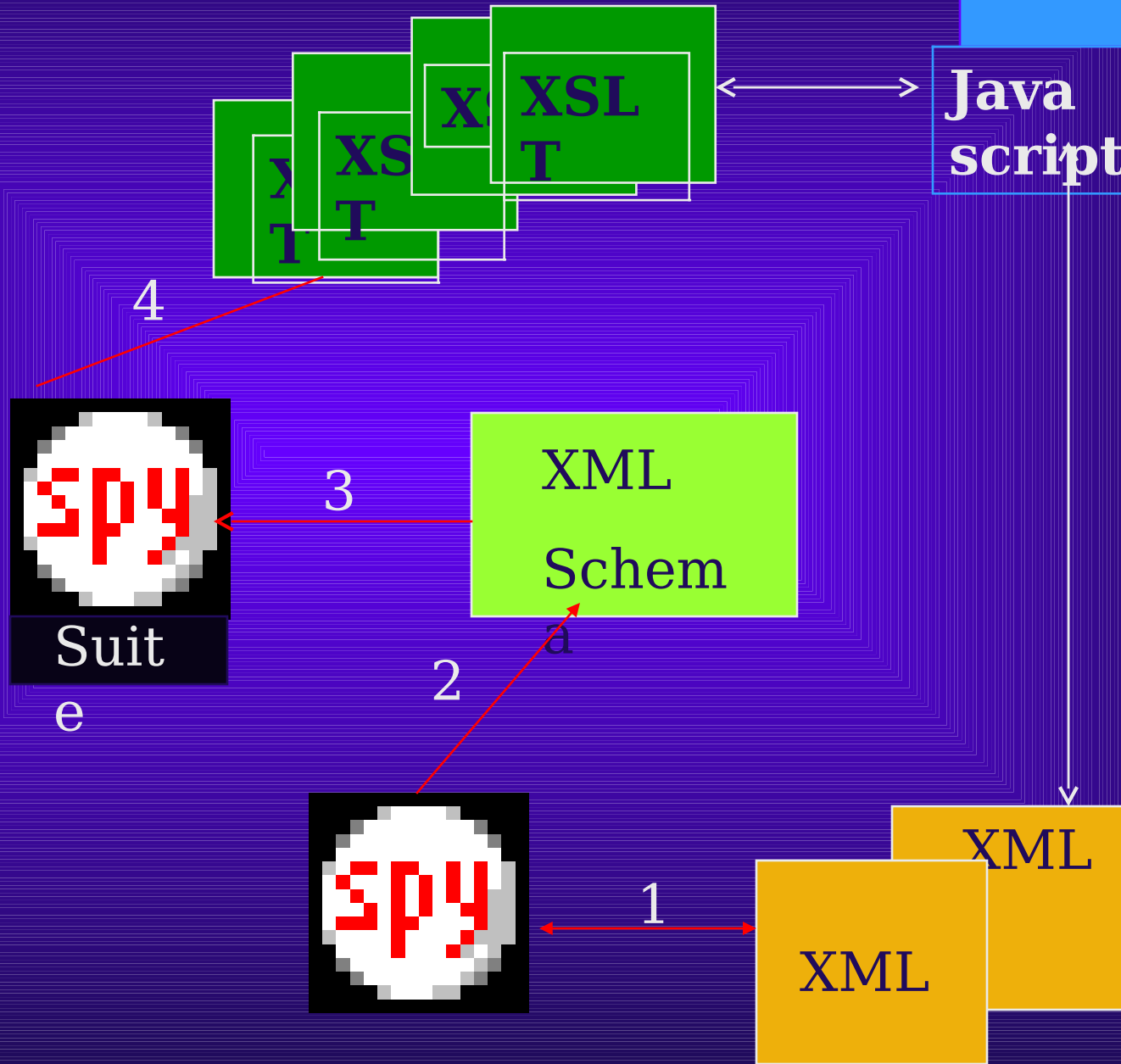
Project's Steps

5 HTML

XSLT
XSLT
XSLT
XSLT

Java script

4

Spy
Suite

3

XML Schema

2

1

Spy

XML
XML

Project's Files Glossary

1- loadXMLFiles.html: Represents the interface in the client side which also have a java script that performs the following functions:

- Creates an instance of XML parser(XMLDOM)

- Loads XML, XSLT files.

- Transforms the XML files using XSLT files and return the result to the browser.

2- MilitaryUnits.xml: Contains information of tanks that belong to the military unit "007".

3- MilitaryUnits.xml: Contains information of tanks that belong to the military unit "101".

4- MilitaryUnits.xsd: Represents the common schema files for XML files

5- MilitaryUnits.xsl: The style sheet file that represents the format of tanks information of military unit "007".

5- MilitaryUnit2.xsl: The style sheet file that represents the format of tanks information of military unit "101".

6- M1A1Tank0.xsl: The style sheet file that represents the format of a part of M1A1 tanks information for any unit chosen by the user.

7- M1A1Tank0.xsl: The style sheet file that represents the format of a part of M1A2

M1a1fire.gif: **Animated tank picture**.

m1.jpg: M**1A1 tank picture**.

m2.jpg: **M1A1 tank picture.**

How to Start Demo?

- Unzip (MV3250-Final Project.zip) file;

- Click on loadXMLFiles.html file.

- **Note:** If you want to get the pictures and the presentation of the project in your web pages you have to change the path of their locations in all the XSLT files and the HTML file.